WHAT IS CLAIMED IS:

1. A method for determining an amount of storage for a level of detail in a MIP map, comprising:

identifying a given level of detail;

identifying a size for an immediately larger level of detail and a magnitude for each dimension of the immediately larger level of detail; and

calculating the amount of storage based on the size and magnitudes without using a multiply operation.

2. The method of claim 1, wherein calculating comprises:

scaling the size.

3. The method of claim 2, wherein scaling comprises:

dividing each of the magnitudes by two and discarding any remainders; and

dividing the size by 2^n and discarding any remainder, where n is the number of non-zero magnitudes remaining after dividing each of the magnitudes;

wherein the size after dividing the size is the amount of storage for the given level of detail.

4. The method of claim 3, further comprising:

adding one to the amount of storage when any of the n least significant bits of the size of the immediately larger level of detail is non-zero.

5. The method of claim 3, wherein dividing each of the magnitudes comprises:

shifting the binary value of the magnitude to the right by one bit.

6. The method of claim 3, wherein dividing the size comprises:

shifting the binary value of the size to the right by n bits.

7. The method of claim 1, wherein a storage alignment restriction requires the starting address for each level of detail to be a multiple of *m* pixels from a predetermined address, wherein identifying a size and magnitudes comprises:

identifying the size and magnitudes in units such that each unit contains *m* pixels.

8. An apparatus for determining an amount of storage for a level of detail in a MIP map, comprising:

means for identifying a given level of detail;

means for identifying a size for an immediately larger level of detail and a magnitude for each dimension of the immediately larger level of detail; and

means for calculating the amount of storage based on the size and magnitudes without using a multiply operation.

9. The apparatus of claim 8, wherein means for calculating comprises:

means for scaling the size.

10. The apparatus of claim 9, wherein means for scaling comprises:

means for dividing each of the magnitudes by two and discarding any remainders; and

means for dividing the size by 2^n and discarding any remainder, where *n* is the number of non-zero magnitudes remaining after dividing each of the magnitudes;

wherein the size after dividing the size is the amount of storage for the given level of detail.

11. The apparatus of claim 10, further comprising:

means for adding one to the amount of storage when any of the *n* least significant bits of the size of the immediately larger level of detail is non-zero.

12. The apparatus of claim 10, wherein means for dividing each of the magnitudes comprises:

means for shifting the binary value of the magnitude to the right by one bit.

13. The apparatus of claim 10, wherein means for dividing the size comprises:

means for shifting the binary value of the size to the right by n bits.

14. The apparatus of claim 8, wherein a storage alignment restriction requires the starting address for each level of detail to be a multiple of m pixels from a predetermined address, wherein means for identifying a size and magnitudes comprises:

means for identifying the size and magnitudes in units such that each unit contains m pixels.

15. A computer program product, tangibly stored on a computer-readable medium, for determining an amount of storage for a level of detail in a MIP map, comprising instructions operable to cause a programmable processor to:

identify a given level of detail;

identify a size for an immediately larger level of detail and a magnitude for each dimension of the immediately larger level of detail; and

calculate the amount of storage based on the size and magnitudes without using a multiply operation.

16. The computer program product of claim 15, wherein instructions operable to cause a programmable processor to calculate comprise instructions operable to cause a programmable processor to:

scale the size.

17. The computer program product of claim 16, wherein instructions operable to cause a programmable processor to scale comprise instructions operable to cause a programmable processor to:

divide each of the magnitudes by two and discarding any remainders; and

divide the size by 2^n and discarding any remainder, where n is the number of non-zero magnitudes remaining after dividing each of the magnitudes;

wherein the size after dividing the size is the amount of storage for the given level of detail.

18. The computer program product of claim 17, further comprising instructions operable to cause a programmable processor to:

add one to the amount of storage when any of the *n* least significant bits of the size of the immediately larger level of detail is non-zero.

19. The computer program product of claim 17, wherein instructions operable to cause a programmable processor to divide each of the magnitudes comprise instructions operable to cause a programmable processor to:

shift the binary value of the magnitude to the right by one bit.

20. The computer program product of claim 17, wherein instructions operable to cause a programmable processor to divide the size comprises:

shift the binary value of the size to the right by *n* bits.

21. The computer program product of claim 15, wherein a storage alignment restriction requires the starting address for each level of detail to be a multiple of *m* pixels from a predetermined address, wherein instructions operable to cause a programmable processor to identify a size and magnitudes comprise instructions operable to cause a programmable processor to:

identify the size and magnitudes in units such that each unit contains *m* pixels.